



Universidad de Granada

## Redes Multiservicio

Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

## Práctica 1

### Configuración de un *router* para la provisión de Calidad de servicio según SLA (6h, 1.5 ptos)

---

#### Objetivos

- Caracterizar el tráfico de aplicaciones multimedia con requisitos de tiempo real.
- Configurar un router para proporcionar QoS a aplicaciones.
- Identificar la configuración de los mecanismos de provisión de QoS para cumplir un acuerdo de nivel de servicio (SLA) básico.

#### Tareas a realizar:

La realización de esta práctica tiene varias partes. Para empezar, se analizará el tráfico generado por varias aplicaciones de red con requisitos de QoS. Después se configurará varios mecanismos de provisión de QoS en un *router* VyOS, para mejorar la calidad de servicio obtenida. Finalmente, se realizará la configuración necesaria para proporcionar la QoS que requiere un SLA básico.

Como resultado, se mostrará su ejecución, y se entregará un informe mostrando como se ha resuelto cada uno de los apartados propuestos.

Las tareas a realizar son las siguientes:

0.- (0.25 puntos) Configure los ordenadores y los *routers* correspondientes para obtener el entorno del laboratorio que se describe en la sección “A.0. Entorno de experimentación”

1.- (0.25 puntos) Caracterice el tráfico que generan dos aplicaciones, mediante el uso del generador de tráfico ITG y el analizador de tráfico *wireshark*. Para ello, siga las especificaciones indicadas en el apartado: “A.1. Caracterización de distintos tipos de tráfico”. La sintaxis básica del generador de tráfico puede consultarla en la sección “B. Sintaxis del generador de tráfico D-ITG”. Debe contestar a las preguntas<sup>1</sup>:

- a) ¿Que tasa de transferencia media y máxima requiere la aplicación Quake3?
- b) ¿Que tasa de transferencia media y máxima requiere la aplicación de VoIP?

1	Sugerencia: ejecutar <i>wireshark</i> en el receptor. Lanzar las aplicaciones con un script.
---	--



Universidad de Granada

**Redes Multiservicio**  
*Grado en Ing. Informática*

**Práctica 1**

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



**Dpt. Teoría de la Señal,  
Telemática y comunicaciones**

- c) (opcional, +0.25 puntos) Exporte los datos a CSV o texto plano, y analice con otra herramienta los retardos entre paquetes y los tamaños de los mismos. Preséntelos como histogramas.
- 2.- (0.25 puntos) Ejecute el generador de paquetes ITG para simular una fuente de tráfico elástico y otra con tráfico inelástico. Configure los *routers* según se especifica en el apartado: “A.2. Tráfico elástico e inelástico”. La sintaxis básica del generador de tráfico puede consultarla en la sección “B. Sintaxis del generador de tráfico D-ITG”. Debe contestar a la siguiente pregunta:
- a) ¿Qué ocurre cuando ambos tipos de tráfico pasan por en el mismo *router* cuando se produce congestión?
- 3.- (0.25 puntos) Genere varios flujos TCP que compitan por el ancho de banda disponible.
- a) ¿Qué ocurre? ¿Cuál obtiene mayor tasa de transmisión?
- 4.- (0.5 puntos) Compare el efecto de configurar en el *router* una política de verificación de conformidad (*policing*) y un esquema de conformado de tráfico (*shaping*) para un mismo flujo de tráfico. Configure los *routers* según se especifica en el apartado: “A.3. Efecto de *policing* y *shaping*”. (Vea además las secciones “C. Configuración de clases de tráfico”, “C.1. Configuración de *policing*” y “C.2. Configuración de *shaper* puro”). Debe contestar a la siguiente cuestión:
- a) ¿Qué diferencia observa en el tráfico tras usar *policing* y *shaping*?
- 5.- (0.5 puntos) Configure el *router* para que ofrezca la QoS requerida por el acuerdo de nivel de servicio (SLA, *Service Level Agreement*) indicado en el apartado: “A.4. Configuración de SLA básico”. Compruebe dicha configuración. Debe especificar razonadamente la configuración que propone y evaluarla.
- 6.- (Opcional, + 0.25 puntos) Genere tráfico de vídeo, y caracterícelo, y calcule los parámetros más restrictivos del *policing* que permita recibir el vídeo con calidad. Para ello, utilice *gstreamer* y *wireshark*: “A.5. Uso básico de *gstreamer*”.



Universidad de Granada

## Redes Multiservicio Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

## A.0. Entorno de experimentación

La práctica debe desarrollarse en el mismo entorno que el creado para el “Seminario 1. Taller de configuración de QoS en routers” según la Figura 1:

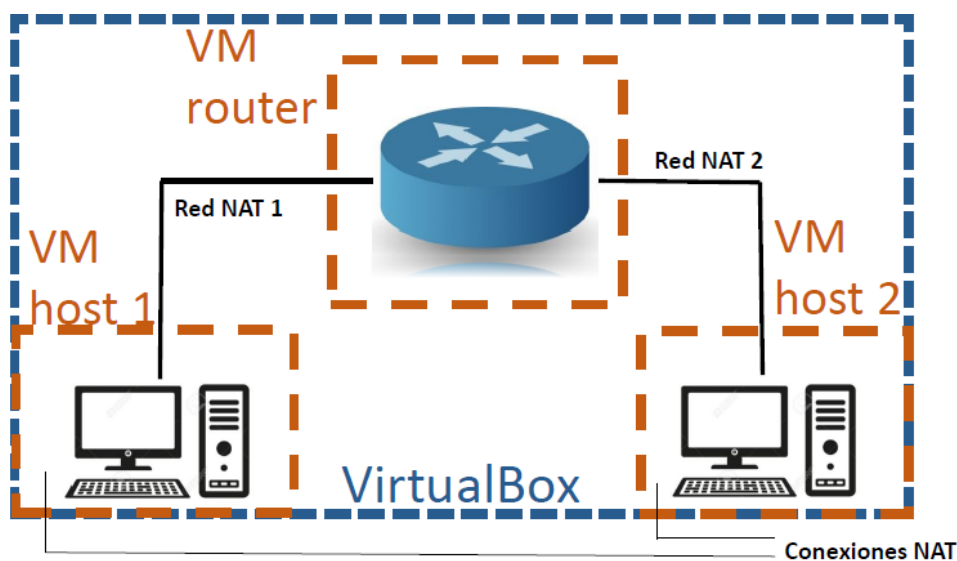


Figura 1. Topología de red para la práctica.

Se vuelven a repetir las direcciones de red del seminario:

Red NAT 1 (10.X.1.0/24)	
Interfaz <i>router</i>	10.X.1.1Y0
Interfaz <i>host</i>	10.X.1.5Y
Red NAT 2 (172.1X.1.0/24)	
Interfaz <i>router</i>	172.1X.1.1Y0
Interfaz <i>host</i>	172.1X.1.2Y

Donde X,Y son el segundo y tercer dígito del DNI, NIE o similar.

Para configurar las interfaces con su dirección IP hay que seguir los pasos propuestos en el seminario. También se debe añadir una ruta por defecto y comprobar la conectividad con `#traceroute`.



Universidad de Granada

## Redes Multiservicio

Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

## A.1. Caracterización de distintos tipos de tráfico

Cada aplicación suele generar un tipo de tráfico que puede caracterizarse. Por lo general, los requisitos de QoS de estas aplicaciones es conocida a priori (p.e., para VoIP con el códec G711, se requiere al menos 64 kbps, y un retardo máximo de 300 ms). Sin embargo, hay otro tipo de tráfico que es necesario analizar, para identificar la tasa de transferencia que requiere, etc. En esta práctica utilizaremos el capturador de *Wireshark*, para caracterizar la tasa de transferencia de dos aplicaciones.

*Wireshark* es un capturador de paquetes, que permite analizar tanto paquetes individuales como flujos completos. Su opción de filtrar tráfico permite seleccionar flujos concretos. Además, puede mostrar en tiempo real la tasa de tráfico recibido en una gráfica. Para ello, arranque la herramienta y tras comenzar a capturar en la interfaz deseada, en "Statistics → I/O Graph".

## A.2. Tráfico elástico e inelástico (VyOS)

Para este ejercicio, hace falta generar un flujo de paquetes UDP y otro TCP, de *bitrate* constante, que compartan el ancho de banda disponible. Para ello:

1.- Limite la tasa que permite el enlace entre el *router* y el segundo ordenador (recepción) a 2Mbps. De esta forma se simula un enlace real con una tasa máxima. Para ello se puede utilizar una política de limitación en la interfaz del *router* dentro de la configuración del mismo:

```
vyos@vyos:~$ configure
vyos@vyos# set traffic policy shaper limitadorenlace default
bandwidth 2mbps
vyos@vyos# set interfaces ethernet eth1 traffic policy out
limitadorenlace
```

2.- Utilice un script con ITGSend con la opción "-T UDP" y "-T TCP", ambos a una tasa de 1200 kbps durante 20 segundos.

3.- Monitorice con *Wireshark* la tasa de transferencia en el receptor.



Universidad de Granada

## Redes Multiservicio

Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

**¡Atención!** Antes de continuar con el siguiente ejercicio no olvide desactivar la limitación de la tasa de transmisión:

```
vyos@vyos:~$ configure
vyos@vyos# delete interfaces ethernet eth1 traffic policy out
limitadorenlace
```

## A.3. Efecto de *policing* y *shaping*

Para este ejercicio es necesario definir para una clase de tráfico una política que use un *policer* y luego un *shaper*, y comparar los resultados de tráfico que generan mediante *wirehark*. Por ello, se aconseja seguir los siguientes pasos:

- 1.- Genere con ITGSend un flujo de tráfico Quake3, con puerto de destino 6666.
- 2.- Defina la clase de tráfico QUAKE3, con el tráfico UDP que vaya dirigido al puerto 6666.
- 3.- Asigne una política de QoS con *police* a 112000 bps y asígnesela a la interfaz Eth0 (*limiter*) o interfaz Eth1 (*shaper*) del *router*. (Consultar “C.1. Configuración de *policing*”)
- 4.- Asigne una política de QoS con *shape* a 112000 bps y asígnesela a la interfaz Eth1 del *router*.

**Nota:** Si ya está asignada la política a un interfaz, al redefinir la política, se aplica el cambio automáticamente en la interfaz (aplicando el cambio de redefinir la política). De esa manera se puede ver instantáneamente el efecto de uno u otro mecanismo.

## A.4. Configuración de SLA básico

Un SLA define el contrato entre cliente y proveedor de servicios. En esta práctica vamos a configurar el *router* para que aplique un SLA que satisfaga las siguientes condiciones:

- El tráfico de voz sobre IP (VoIP) tendrá prioridad, con 80 kbps garantizados.
- El tráfico de Quake3 estará conformado a 112 kbps.
- El resto del tráfico se debe limitar a 2 Mbps. Cualquier tráfico que sobre pase estos umbrales debe ser descartado.



Universidad de Granada

**Redes Multiservicio**  
Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

Para probarlo, generar el siguiente patrón de tráfico, y monitorice el tráfico que se recibe:

- Genere un flujo de VoIP.
- Genere 3 flujos diferentes de tráfico UDP a distintos puertos, con una tasa de 800 kbps cada uno.
- Genere un flujo de tráfico Quake3.

## A.5. Configuración básica de *gststreamer*

Para generar tráfico de vídeo, se puede utilizar la librería *gststreamer* [GSTREAMER] para capturar vídeo y enviarlo mediante con el protocolo RTP. Para ello, si deseamos codificar el vídeo con el códec h.264 al puerto 554, en el receptor (con IP=10.6.2.2) se puede usar la siguiente línea:

```
# gst-launch-0.10 udpsrc port=554 ! "application/x-  
rtp,payload=127" ! rtph264depay ! ffdec_h264 ! xvimagesink  
sync=false
```

Por otro lado, en el emisor hace falta ejecutar la siguiente línea:

```
# gst-launch-0.10 v4l2src device=/dev/video0 ! 'video/x-raw-  
yuv,width=320,height=240' ! x264enc pass=qual quantizer=20  
tune=zerolatency ! rtph264pay ! udpsink host=10.6.2.2 port=554
```

## B. Sintaxis del generador de tráfico D-ITG

La herramienta D-ITG (*Distributed Internet Traffic Generator*, [ITG12]), permite generar tráfico de red de forma configurable. Además, incluye modelos que le permiten inyectar el tráfico que generan aplicaciones conocidas. El funcionamiento de ITG requiere lanzar la aplicación receptora (ITGRecv) en el ordenador donde va destinado el tráfico, con la siguiente sintaxis:

```
# ./ITGRecv
```



Universidad de Granada

## Redes Multiservicio

Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

Además, es necesario lanzar la parte emisora de tráfico (ITGSend), a la que se le puede indicar distintos parámetros que regulan el tráfico que se generará. La sintaxis básica de la herramienta es:

```
# ./ITGSend -a <dirección-IP-destino> -rp <puerto-destino> -t  
<duración-de-envío-de-paquetes-en-ms> -l <fichero-con-  
resultados> -C <tasa-de-paquetes-por-segundo> -c <tamaño-de-  
paquete> -T <protocolo-a-generar: UDP, TCP, SCTP...> -d  
<retraso-inicial> -z <número-de-paquetes>
```

Con la sintaxis anterior podemos generar tráfico sintético con una tasa de bits constante. No obstante, es posible generar tráfico más realista indicando que siga distribuciones de periodos entre paquetes y tamaños de paquetes (ejecutar `./ITGSend -help` para más información).

Para generar tráfico de aplicaciones conocidas (como Quake3 o VoIP), podemos seguir la siguiente sintaxis:

```
# ./ITGSend -a <dirección-IP-destino> -rp <puerto-destino> -t  
<duración-de-envío-de-paquetes-en-ms> -l <fichero-con-  
resultados> <Aplicación: Quake3, VoIP, CSa, etc.>
```

Tras la ejecución de la herramienta, los resultados pueden examinarse mediante la herramienta ITGDec:

```
# ./ITGDec <fichero-con-resultados>
```

Si necesitamos generar varios flujos de tráfico, podemos crear un script con la configuración de cada flujo a generar. Dicho script consiste en una secuencia de línea con la sintaxis de especificación de tráfico descrita. Por ejemplo, con las siguientes líneas se generaría durante 10 s tráfico de voz sobre IP y del juego Quake3:

```
# cat > script.txt  
-a localhost -rp 3000 -t 20000 VoIP  
-a localhost -rp 3001 -t 20000 Quake3  
  
# ./ITGSend script.txt -l script.log
```



Universidad de Granada

**Redes Multiservicio**  
Grado en Ing. Informática

### Práctica 1

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



Dpt. Teoría de la Señal,  
Telemática y comunicaciones

## C. Configuración de reglas de tráfico (VyOS)

El primer paso para hacer un trato diferenciado de tráfico, es la clasificación de paquetes. Esta clasificación puede llevarse a cabo considerando campos de los protocolos de distintas capas de red. Dependiendo del dispositivo que se utilice, los campos seleccionables son diferentes. Hay que tener en cuenta que la clasificación puede ser un proceso costoso, si se utilizan campos de protocolos complejos.

*En IOS de CISCO es posible realizar la clasificación mediante “class-map” y “access-list”.*

En VyOS se utilizan directamente políticas. Para usarlas es necesario crear un *traffic-policy* con una política concreta. Tras ello se define la clase (class) por defecto y después diversas clases en particular. Para cada clase se definen una serie de parámetros (p. e. *bandwidth*) y el filtro *match* (al que se le puede asignar un protocolo concreto, puerto...). Tras ellos siempre se debe asignar a la interfaz para registrar la política en tráfico entrante o saliente.

**Nota:** La política *shaper* es una política genérica (para *outbound traffic*) que no significa que se haga un *shaping* como tal. Simplemente permite modificar de forma genérica el tráfico de salida con unos atributos particulares.

### C.1. Configuración de *policing* (VyOS)

La verificación de la conformación de tráfico (*policing*) consiste en comprobar que el tráfico de paquetes cumple una especificación de tráfico. Si supera los límites definidos en los patrones, se lleva a cabo una acción sobre el paquete recibido. Estas acciones suelen ser descartar el paquete o marcarlo. El *policing* se suele ejecutar a la entrada del *router* de acceso a la red. Así se puede controlar el flujo de tráfico que llega a una red, y predecir cómo se va a comportar la red, y qué QoS puede ofrecer (retardo, ancho de banda, paquetes perdidos, *jitter*...). También se puede implementar a la salida.

Para definir un limitador como tal se debe usar la política *limiter* y aplicarla en la interfaz de entrada (eth0) para el tráfico entrante al *router* (*inbound traffic*). También existe una forma menos simple que es creando un *shaper* sin cola de forma que la política sería descartar los paquetes para el tráfico saliente de la interfaz (eth1, *outbound traffic*).





Universidad de Granada

**Redes Multiservicio**  
*Grado en Ing. Informática*

**Práctica 1**

Configuración de un *router* para la provisión de  
Calidad de servicio según SLA



**Dpt. Teoría de la Señal,  
Telemática y comunicaciones**

## C.2. Configuración del *shaper* puro (VyOS)

Otro mecanismo de provisión de QoS, que permite modificar el patrón de tráfico de salida de un *router* para que los paquetes sigan un tráfico regular, sin picos, es el *traffic shaping*, o conformación de tráfico. El funcionamiento del *traffic shaper* se basa en el modelo de *token-bucket*. La configuración es similar a la del *policing*.

Se debe aplicar la política *rate-control* (para el tráfico saliente, *outbound traffic* en el eth1). Este es el filtro auténtico de cubo horadado, también existen otros conformadores.

## Bibliografía básica y complementaria:

[ETHEREAL] [WIRESHARK] "WireShark. User's guide.", <http://www.wireshark.org/docs/>

[ITG12] A. Botta, A. Dainotti, A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios", *Computer Networks* (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547.

[VyOS Traffic Policy],  
<https://docs.vyos.io/en/latest/configuration/trafficpolicy/index.html>